Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Rapid Control Prototyping for Robot Soccer ⋆

Junwon Jang * **Soohee Han** ** **Hanjun Kim** * **Choon Ki Ahn** *

* *School of Electrical Engr. & Computer Science, Seoul National University, Korea (e-mail: jwchang@cisl.snu.ac.kr).*
** *BK21 School for Creative Engineering Design of Next Generation Mechanical and Aerospace Systems, Seoul National University, Korea (e-mail: hsh@cisl.snu.ac.kr)*

Abstract: In this paper, we propose rapid control prototyping (RCP) for a robot soccer using the SIMTool that has been developed in Seoul National University, Korea for the control aided control system design (CACSD). The proposed RCP enables us to carry out the rapid design and the verification of controls for two-wheeled mobile robots (TWMRs), players in the robot soccer, without writing C codes directly and requiring a special H/W. On the basis of the proposed RCP, a blockset for the robot soccer is developed for easy design of a variety of mathematical and logical algorithms. All blocks in the blockset are made up of basic blocks offered by the SIMTool. User-defined algorithms can be easily and efficiently constructed with just a combination of the blocks in the blockset. In order to validate the proposed RCP in a real game, we employ an official simulation game for the robot soccer, the SimuroSot. Block diagrams are constructed for strategy, path calculation, and the interface to the SIMTool. We show that the algorithms implemented with the proposed RCP work well in the simulation game.

1. INTRODUCTION

Recently, model-based design has been considered as an efficient and reliable method for the development of the controls since the performance can be easily verified in advance and even improved through iterative processes between simulations and experiments. Beginning in the automotive field, model based design has been extended to a variety of areas such as process controls, defense, industrial equipment, and so on. As an important tool of model-based design, rapid control prototyping (RCP) has been widely used, which provides a method of easily verifying the designed control on a real system and monitoring the result on a real-time basis.

As the advancement of robot technologies, robot soccer becomes popular over the world. Robot soccer systems have been developed with high technologies of H/W. Accordingly, the robot soccer systems need efficient algorithms to manage H/W in a proper manner and then achieve the good performance. In order to develop such efficient algorithms, much knowledge on various algorithms is often required. Furthermore, it may also take a lot of time and efforts to write hard C codes for implementation. Our motivation comes from the belief that the RCP for robot soccer systems can be very helpful for the beginners who are unfamiliar with robot soccer systems and for the experts who want to design the prototypes of the novel algorithms rapidly and evaluate them.

In order to build the development environment supporting the RCP for robot soccer systems, computer aided control systems design (CACSD) tools are required, which should be able to design the controls for two-wheeled mobile robots (TWMRs) operating in the robot soccer system. With the CACSD tools, we should be able to efficiently develop and validate lots of controllers for TWMRs even with nonholonomic constraints. In addition, the prototypes of the algorithms for robot soccer system such as shoot, obstacle avoidance, goalkeeping, and so on, can be also developed fast without writing C codes directly. As a CACSD tool for the RCP, we employ the SIMTool that has been developed in Seoul National University, Korea [3].

In this paper, we propose the RCP for TWMRs using the SIMTool. Under the proposed RCP, a robot soccer blockset is developed for easy design of the algorithms. We also validate the proposed RCP by applying it to a simulation game for the robot soccer.

The proposed RCP with the SIMTool for TWMRs enables us to carry out the rapid design and verification of the controllers or algorithms for TWMRs in the robot soccer system without writing C codes directly. All the functions in the SIMTool can be utilized, which includes setting various parameters for experiments easily, acquiring data, drawing the graphs and so on.

By using the robot soccer blockset with the SIMTool that is developed on the basis of the above RCP, the simple functions for the robot soccer are easily implemented with several basic blocks offered by the SIMTool. A rather complex function for the robot soccer can be designed rapidly and efficiently if common and repetitive functions are already implemented with special blocks. A set of such special blocks are often called a blockset.

⋆ This work was supported by the second stage of the Brain Korea 21 Project in 2007.

In this paper, we validate the proposed RCP by implementing algorithms for strategy, path calculation, and the interface to the SIMTool, and applying them to a simulation game for the robot soccer. Since it is often troublesome to have all equipment for the robot soccer systems, the proposed RCP with a simulation game would provide a convenient development environment and hence save costs and effort.

This paper is organized as follows. In Section 2, we introduce the robot soccer system and the concept of the RCP. We perform simulations and experiments for the validation of the proposed approach in Section 3. In Sections 4, we validate the proposed RCP with a simulation game, the SimuroSot. The conclusions are drawn in Section 5.

2. BACKGROUND

In this section, we briefly introduce a basic knowledge on the robot soccer and the RCP. The requirements for them are also discussed.

2.1 Robot soccer system

Overview The robot soccer started in 1980s. At present two international soccer groups, Robocup and FIRA, hold two kinds of matches for computer simulations and real games on the play ground. Among real games, MIROSOT (Micro Robot World Cup Soccer Tournament) competition is most popular and has the following rule: The teams consisting of 3, 5, or 11 robots play on the playground of size 150 cm × 130 cm, 220 cm × 180 cm, or 440 cm × 280cm, respectively [4]. TWMRs, players in the robot soccer, should be smaller than cubes with sides 7.5 cm long and have two wheels driven by motors. A microcontroller inside the TWMR regulates the toque of the wheels according to commands transferred from PC over the wireless communication. A CCD camera and an RF antenna over the playground send the captured images to the PC and transfer commands generated by programmed strategy on PC to its own TWMR via wireless communication, respectively. Two teams have their own CCD cameras, RF antennas, and PCs. In the similar way to a real soccer game, the referee manages the game. At the beginning of the game and at every interruption, TWMRs have to be placed in their specific positions according to the rules in respect of the starting and interruption cases. The teams start the programs on their PC immediately after the referee blows a whistle. TWMRs should play autonomously until the interruption happens. During the game any interference due to human intervention is prohibited and penalized.

As seen in devices composing robot soccer systems, the developments of the robot soccer system cover a wide range of areas such as robotics, sensor fusion, control and estimation, communication, image processing, computer technology, and so on. So, much knowledge and efforts are required to implement S/W such as algorithms together with H/W for the robot soccer system. Even a simple algorithm may take a lot of efforts and a long time to be evaluated on a real system. The RPC discussed in this paper can give much help to save such efforts and time, and hence make them concentrating on the development of algorithms instead of implementation.

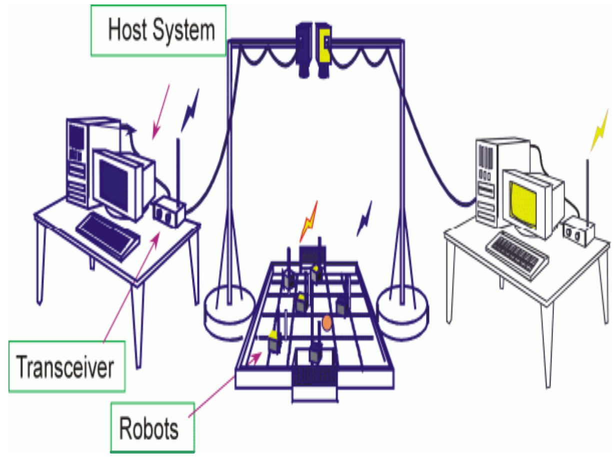

Fig. 1. The overall view of a robot soccer system [4]

System configuration The robot soccer system is composed of several components.

As mentioned before, the robot soccer system has TWMRs as players. Each TWMR must operate with fully independent powering and motoring mechanisms. TWMRs have their own on-board processor so that they can communicate with a host computer.

A vision system consists of a color-CCD-camera and a frame grabber. The vision system delivers the position of TWMRs and the ball on the playground. In a 3 to 3 match, it should distinguish at least six colors, *i.e.*, two colors for two teams (own and opponent), three for the TWMRs and one for the ball. Each robot puts on a uniform to identify a team and itself.

A host computer generally performs the tasks for generating commands to TWMRs. The positions of the ball and each robot are obtained from the vision system and the commands for moving directions of TWMRs are determined on the basis of the strategy and the path. In some cases, the host computer manages the communication between players.

For the wireless communication between TMWRs and a host computer, half-duplex RF-Modules (Frequency: 418 MHz or 433 MHz) are used as a standard.

Additionally, we need the S/W that has the following functionalities:

- Image processing (vision system)
- Communication
- Game strategy

The software delivering information from vision systems may be available from vendors. In this paper, we employ the software provided by vendors and do not consider game strategy.

2.2 Rapid Control Prototyping

The RCP helps the engineers quickly implement and evaluate their control strategies on a real system with input/output devices. The RCP differs from the hardware-in-the-loop (HIL) simulation in that the former applies a simulated control to a real system while the latter applies a real control to a simulated system [5]. Since the controls in the RCP are easily implemented and adjusted, the RCP is more suitable for control designs than the HIL simulation.

The typical RCP system is generally composed of the following components:

- Basic math functions.
- A program translating a block diagram to the corresponding C codes.
- A symbolic I/O blockset for a CACSD tool.
- A real-time target computer with I/O interfaces to analog and digital signals.
- A host PC communicating with a target computer.
- A Graphical user interface (GUI) application monitoring the real-time process.

All algorithms in the RCP are developed with symbolic models, not with C-codes, and they are designed using a popular math modelling software. We can concentrate on developing our algorithms in a visual and intuitive modeling environment and do not have to worry about translating the model to C-code. This is very significant since most control engineers are not C-code experts nor do they typically have the skills to port C-code to a real-time target. By virtue of an automated build process, the RCP system gives a convenient development environment to control engineers.

The RCPs fall into two categories, DSP-based rapid prototyping and PC-based rapid prototyping. DSP-based rapid prototyping is so fast that it has been widely used in industries despite of the high cost. PC-based rapid prototyping, however, has the cost efficiency and the ability of extensions to other systems, which is replacing the DSP-based prototyping system with the improvement of the components such as the CPU and the memory. These days PC-based prototyping system is getting preferable. In this paper, we also employ the PC-based RCP.

3. SIMULATION AND EXPERIMENT

In order to verify the applicability of the proposed RCP environment with the provided blockset, both simulation and experiment have been performed and their results are compared.

3.1 Experimental configuration

To illustrate the proposed RCP environment, an experimental testbed is constructed as seen in Fig. 2. The SIMTool and a vision application program run on the windows systems of a host computer that is equipped with CPU 2.8 GHz. A TWMR is made in Microrobot Inc., which has 7.5 cm × 7.5 cm × 7.5 cm in size and the maximum velocity 1.5 m/s. The TWMR has the CPU, an Am188ES micro processor running at 40MHz, and DC motors with L298 driver chips. Besides, the RS232c serial communication is used to receive the command input from the host computer. For vision system, a CCD camera with 640 × 480 pixels and MyVision Image Grabber made in Microrobot Inc. with 30 frames/s in PCI slot are employed.

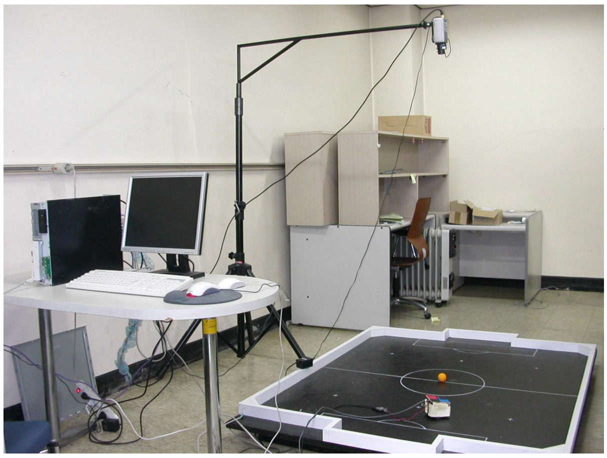

Fig. 2. The overall view of the robot soccer system for experiments

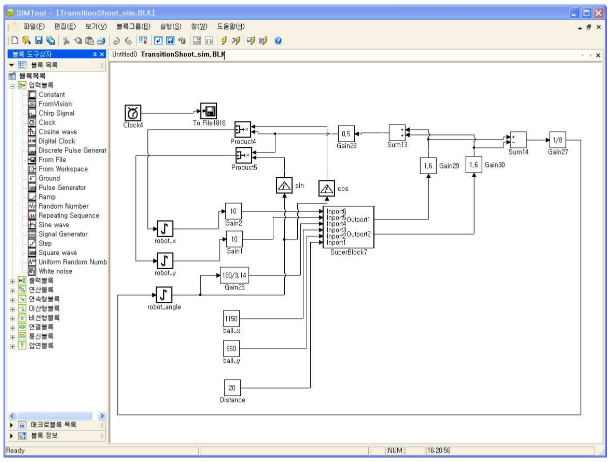

Fig. 3. A SIMTool block diagram model for the transition shoot

3.2 Results of simulations and experiments

Here, we take three examples, transition shoot, turning shoot, and the obstacle avoidance to show that the algorithms designed with the RCP work well.

Fig. 3 shows a block diagram for transition shoot that includes a large *Transition Shoot* block in the center. Fig. 4 shows the internal view of *Transition Shoot* block in the SIMTool. While Fig. 3 and 4 are for simulation, Fig. 5 is for experiment with the RCP environment. Transition shoot is done by the following steps: The proper position for transition shoot is first chosen. A TWMR turns around toward that position and go forward. When the TWMR gets there, it turns around toward the ball again. Finally it runs to the ball to shoot it to the goal. Fig. 6 and 7 show the trajectories of a TWMR from simulation and experiment, respectively. The TWMR and the ball are initially placed in (x = 38 cm, y = 30 cm, θ = 0 rad) and (x = 115 cm, y = 65 cm), respectively. The ball is fixed before shooting. The simulation time and its corresponding sampling time are set to 10 seconds and 0.01 seconds. The experiment time and its corresponding sampling time are set to 20 seconds and 0.01 seconds. It is observed that two trajectories from simulation and experiment are very similar to each other. As another

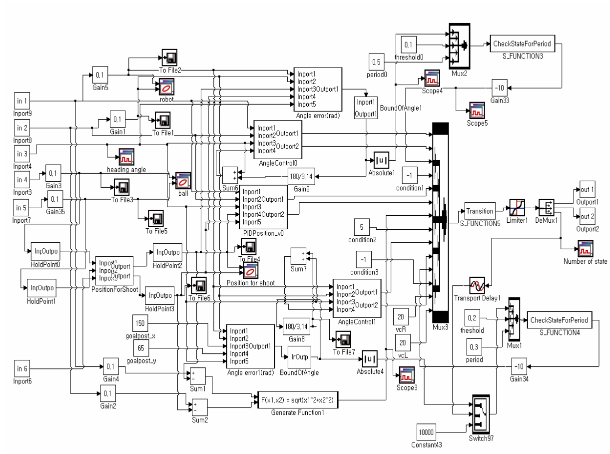

Fig. 4. Internal view of the *Transition Shoot* block

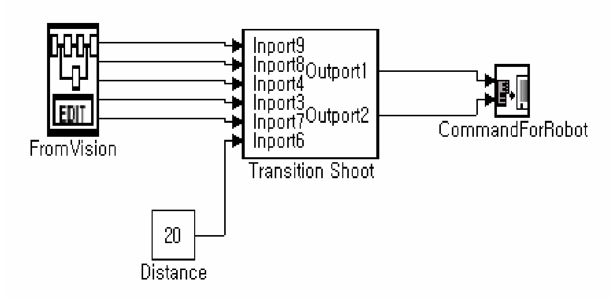

Fig. 5. A SIMTool block diagram for the experiment of the transition shoot

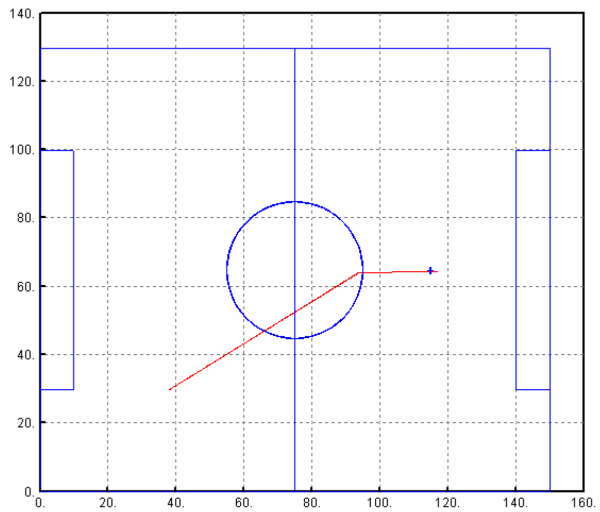

Fig. 6. The simulated trajectory of a TWMR for the transition shoot

example, we now consider the turning shoot. Fig. 8 and 9 show the trajectories of a TWMR from simulation and experiment, respectively.

In order to do a turning shoot, a turning point is first chosen by considering the positions of the ball and the goal. A TWMR moves to the turning point and then goes around in circle to shoot the ball. The TWMR and the ball are initially placed in (x = 124 cm, y = 26 cm, θ = 1.57 rad) and (x = 75 cm, y = 65 cm),

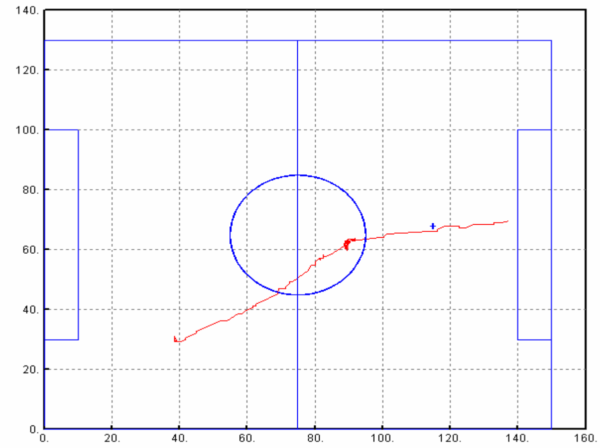

Fig. 7. The real trajectory of a TWMR for the transition shoot

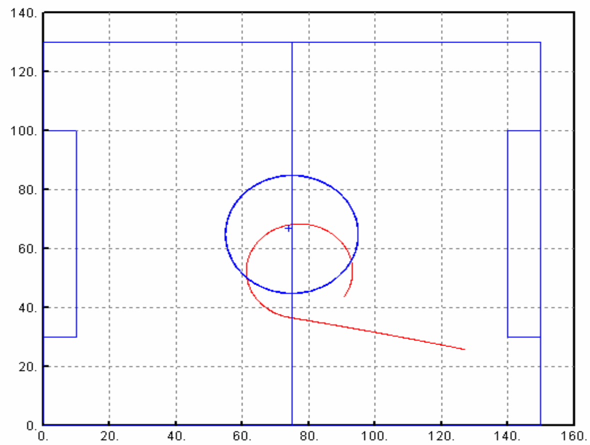

Fig. 8. The simulated trajectory of a TWMR for the turning shoot

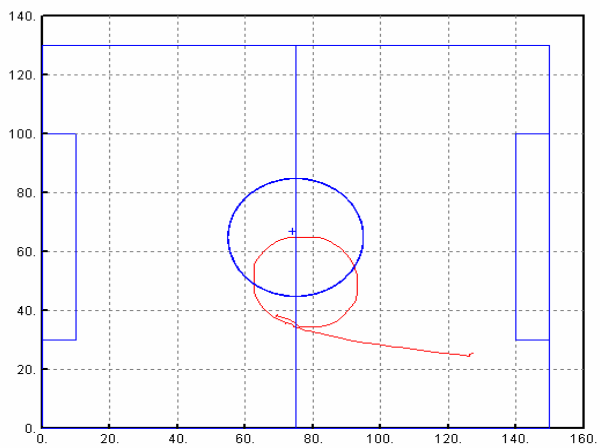

Fig. 9. The real trajectory of a TWMR for the turning shoot

respectively. As in the transition shooing, the ball is fixed before shooting. Here, the TWMR is set to go around in a circle of a 16 cm radius. The simulation time and its corresponding sampling time are chosen to be 15 seconds and 0.01 seconds. The experiment time and its corresponding sampling time are set to 30 seconds and

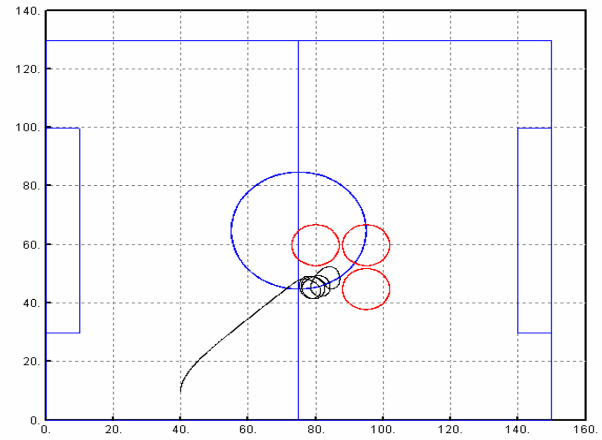

Fig. 10. Simulated local minima in three obstacles when the APF is used.

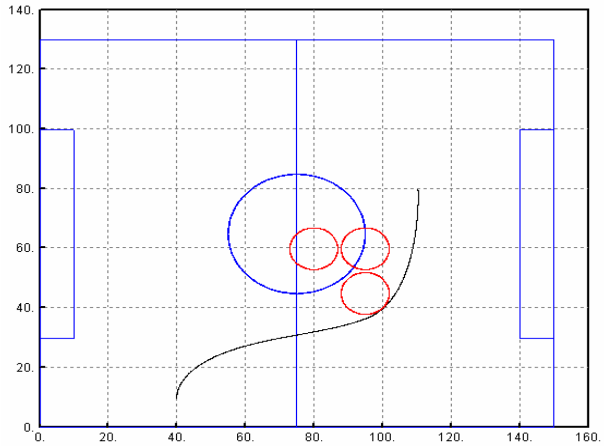

Fig. 11. Simulated avoidance of local minima in three obstacles when the RNDAVS is used.

0.01 seconds. It is observed that two trajectories from simulation and experiment are very similar to each other.

4. APPLICATION TO SIMUROSOT GAME

By carrying out the simulations and the experiments for simple movements of TWMRs in the previous section, we validated the effectiveness of the algorithms designed with the proposed RCP. Here, we apply the proposed RCP to the SimuroSot which is an official simulated robot soccer game approved by FIRA. Since the SimuroSot models a most popular competition, MicroSot, it creates an extremely realistic experience of playing robot soccer on the field.

The SimuroSot consists of a server program and two client ones. The former manages the soccer game environments such as playground, physical behaviors of TWMRs, and a score board. The latter is in charge of the game strategy that keeps TWMRs of each team in good working order. Each team can make their own strategies through their client program and compete with the other team on the simulation program of the SimuroSot[4]. When we would like to evaluate the developed algorithm, the other team can be operated by the SimuroSot. The SimuroSot

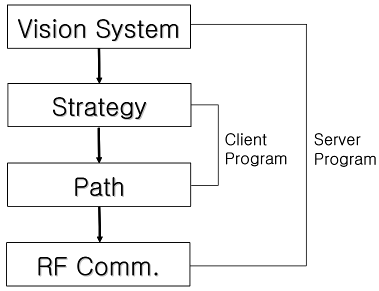

Fig. 12. Server and client programs for a simulated robot soccer game

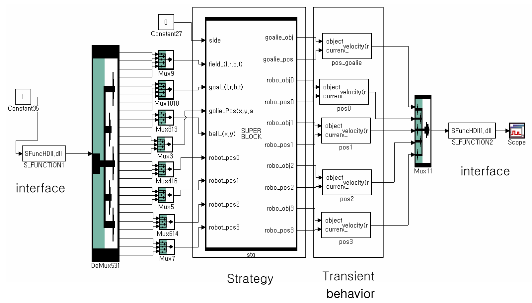

Fig. 13. A block diagram for the client program

supports 5 vs. 5 and 11 vs. 11 plays. In this paper, we choose the 5 vs. 5 play.

4.1 Client programs for determining the overall strategy and the transient behavior

Fig. 12 shows how the process of a robot soccer program runs. Since a server program offers virtual environments replacing a vision system and equipments for wireless communication, we have only to develop algorithms for the overall strategy and the transient behavior to adopt. Fig. 13 shows a block diagram for the client program that describes the overall strategy and determines the transient behavior. If we go into one of blocks outlined in Fig. 13, we can see a block diagram in Fig. 14.

A block diagram for the overall strategy
Four main blocks in Fig. 14 are in charge of the strategy algorithms such as goalkeeping, assigning tasks, positioning TWMRs, and organizing TWMRs. The block for goalkeeping determines the goalkeeper's position so that the ball from the attack of the other team does not get into the goal. The block for assigning tasks puts the remaining robots on their jobs according to their current positions and the distance to the ball. The algorithm employed in this paper has a policy that two TWMRs closest to the ball are appointed as attackers and the others as defenders. The block for positioning calculates the target positions in consideration of TWMRs' current positions and their roles. Each TWMR takes one target position through the block for organizing players

Interface between the SIMTool and the SimuroSot In order to communicate between the SIMTool and the SimuroSot, it is necessary to make an interface which connects them. In the SIMTool, external programs with the *C* language can be imported through an S-function block in a form of a dll file. The server program in the

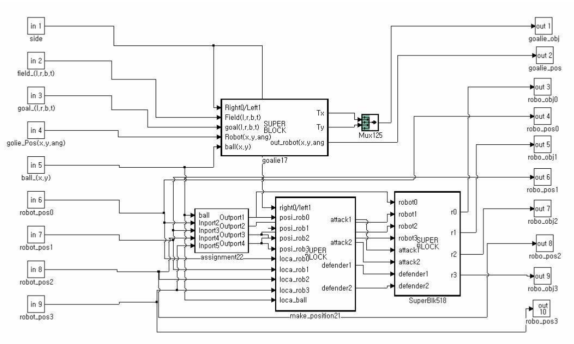

Fig. 14. A block diagram for the overall strategy

Fig. 15. The SimuroSot game : One team is denoted with a circle. One goalkeeper stands in front of the goal line. Two defenders stay in the half line. Two attackers get near the goal area of the other team.

SimuroSot also takes a dll file to import external programs for strategies. Two dll files linked to the CEMTool and the SimuroSot utilize the shared memory to communicate with each other. The shared memory is created by one of two programs and then two programs can transfer data through it as a medium.

The aforementioned client program and its interfaces to the CEMTool are applied to a real game in what follows.

4.2 Simulation with SimuroSot

Now, we play the SimuroSot game based on the proposed RCP. Fig. 15 shows that two teams of 10 TWMRs compete for the goal. Five TWMRs denoted by circles are moved according to the algorithms based on the proposed RCP. The remaining five TWMRs of the other team follow internal algorithms offered by the SimuroSot. As mentioned before, the algorithm employed in this paper has a policy that two TWMRs follow and shoot the ball as attackers, and two TWMRs as defenders stay around our side to keep the other team from getting the goal.

5. CONCLUSION

In this paper, rapid control prototyping (RCP) for a robot soccer was proposed with the SIMTool that has been developed in Seoul National University, Korea for the control aided control system design (CACSD). The proposed RCP is effective in carrying out the rapid design and the verification of controls for two-wheeled mobile robots (TWMRs), players in the robot soccer, without writing C codes directly and requiring a special H/W. It turned out that, on the basis of the proposed RCP, user-defined algorithms can be easily and efficiently implemented by putting together several blocks of the blockset. In order to validate the proposed RCP in a real game, we employed an official simulation game for the robot soccer, the SimuroSot. Block diagrams were constructed for strategy, path calculation, and the interface to the SIMTool. We showed that the algorithms implemented with the proposed RCP work well in the simulation game.

The proposed RCP approach with the developed blockset in this paper has much room for extension and improvement. If further elaboration is applied, the RCP in this paper could give much help for designing efficient algorithms of TWMRs in robot soccer systems.

REFERENCES

[1] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation,", in Proc, IEEE Conf. Robot. Automat, Sacramento, CA, pp. 1398-1404, Apr. 7-12, 1991.
[2] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Robot. Automat.*, vol. 15, no. 5, pp 615-620, 2000.
[3] http://cisl.snu.ac.kr
[4] http://www.fira.net
[5] http://www.precisionmba.com
[6] J. H. Kim, D. H. Kim, Y. J Kim, and K. T Seow, *Soccer Robotics*. Springer Tracts in Advanced Robotics, 2004.
[7] X. J. Motai and Y. X. Zhu, "Predictive fuzzy control for a mobile robot with nonholonomic constraints," Advanced Robotics, ICAR '05. Proceedings., 12th International Conference, pp. 58-65, 2005.
[8] K. C. Park, H. K. Chung, and J. G Lee, "Point Stabilization of Mobile Robots via State-Space Exact Feedback Linearization, "Robotics and Computer Integraged Manufacturing," vol. 16, pp. 353-363, 2000.
[9] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for a non-holonomic mobile robot," IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS'91., Osaka, Japan, pp. 1236-1241, Nov. 3-5, 1991.
[10] H. S. Shim, "Design and implementation of multi-robot cooperation system using BIOS : Its application to robot soccer," Ph.D. dissertation, Dep. of Electrical Eng. KAIST, Taejon-shi, South Korea, 1998.
[11] O. Khatib, "Real-time obstacle avoidance for manipulator and mobile robots," Int. J. Robot. Res., vol. 5, no. 1, pp. 90-98, 1986.
[12] Y. J. Kim, J. H. Kim, and D. S. Kwon, "Evolutionary programming based univector field navigation method for fast mobile robots," IEEE Trans. Syst. Man. Cybern., vol. 31, no. 3, pp. 450-458, 2001.
[13] H. S. Shim, H. S. Kim, M. J. Jung, I. H. Choi, J. H. Kim, and J. O Kim, "Designing Distributed Control Architecture for Cooperative Multi-agent System and Its Real-time Application to Soccer Robot," Robotics and Autonomous Systems, vol. 21, pp. 149-165, 1997.
[14] D. H. Kim and J. H. Kim, "A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer," Robotics and Autonomous Systems, vol. 42, pp. 17-30, 2003